(19) Europäisches Pat ntamt
European Patent Offic
Office ur péen des brevets

(11) EP 1 006 716 A2

(12) EUROPEAN PATENT APPLICATION

(43) Date of publication:
07.06.2000 Bulletin 2000/23

(51) Int Cl.7: H04N 1/64, H04N 1/41, G06T 5/00

(21) Application number: 99309522.3

(22) Date of filing: 29.11.1999

(84) Designated Contracting States:
AT BE CH CY DE DK ES FI FR GB GR IE IT LI LU MC NL PT SE
Designated Extension States:
AL LT LV MK RO SI

(30) Priority: 02.12.1998 US 203870

(71) Applicant: Xerox Corporation
Rochester, New York 14644 (US)

(72) Inventors:
- Fan, Zhigang
Webster, NY 14580 (US)
- Xu, Ming
Rochester, NY 14618 (US)

(74) Representative: Skone James, Robert Edmund
GILL JENNINGS & EVERY
Broadgate House
7 Eldon Street
London EC2M 7LH (GB)

(54) Method and apparatus for segmenting data to create mixed raster content planes

(57) An improved technique for compressing a color or gray scale pixel map representing a document using an MRC format includes a method of segmenting an original pixel map into two planes (12,16), and then compressing the data or each plane in an efficient manner. The image is segmented by separating the image into two portions at the edges. One plane contains image data for the dark sides of the edges, while image data for the bright sides of the edges and the smooth portions of the image are placed on the other plane. This results in improved image compression ratios and enhanced image quality.

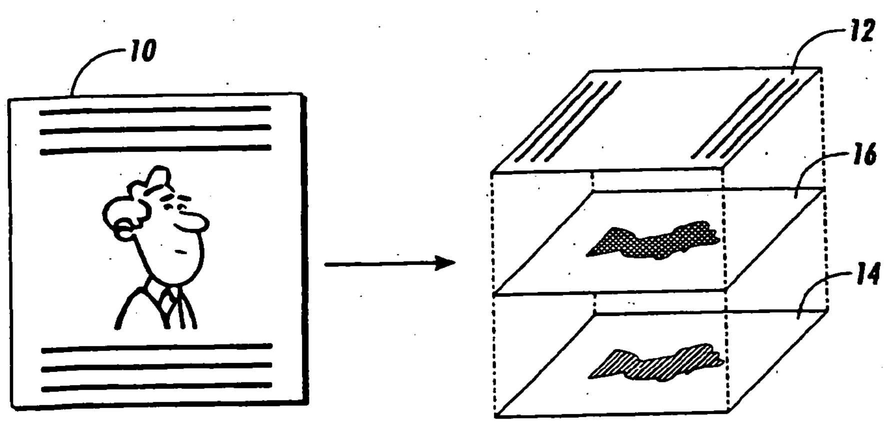

FIG. 1

EP 1 006 716 A2

Des ription

[0001] This invention relates generally to image processing and, more particularly, to techniques for segmenting, classifying and/or compressing the digital representation of a document.

[0002] Documents scanned at high resolutions require very large amounts of storage space. Instead of being stored as is, the data is typically subjected to some form of data compression in order to reduce its volume, and thereby avoid the high costs associated with storing it. "Lossless" compression methods such as Lempel-Ziv Welch (LZW) do not perform particularly well on scanned pixel maps. While "lossy" methods such as JPEG work fairly well on continuous-tone pixel maps, they do not work particularly well on the parts of the page that contain text. To optimize image data compression, techniques, which can recognize the type of data being compressed, are needed.

[0003] Known compression techniques are described in US-A-5778092, US-A-5251271, US-A-5060980, US-A-5784175, US-A-5303313 and US-A-5432870.

[0004] In one embodiment, the present invention discloses a method of segmenting a pixel map representation of a document which includes the steps of: acquiring a block of the digital image data, wherein the digital image data is composed of light intensity signals in discrete locations; designating a classification for the block and providing an indication about a context of the block; segmenting the light intensity signals in the block into an upper subset and a lower subset based upon the designated classification; generating a selector set which tracks the light intensity segmentation; and separately compressing the digital image data contained in the upper and lower subsets.

[0005] In another embodiment, the present invention discloses a method of classifying a block of digital image data into one of a plurality of image data types, wherein the block of data is composed of light intensity signals in discrete locations, which includes: dividing the block into a bright region and a dark region; dividing a low pass filtered version of the block into a bright region and a dark region; calculating average light intensity values for each of the bright region, the dark region, the filtered bright region and the filtered dark region; and comparing a difference between the bright region and the dark region average light intensity values to a filtered difference between the bright region and the dark region average filtered light intensity values; if the average light intensity difference and the average filtered light intensity differ- nce are approximately equal finding a range of values in which the difference value falls, and classifying the block based upon the value rang ; and if the av rage light int nsity differenc and the av rage filter d light intensity difference are not approximately equal finding a range of values in which the filt red difference value falls and classifying the block based upon the filtered value range.

[0006] Some examples of methods according to th present invention will now be described with reference to the accompanying drawings, in which:-

Figure 1 illustrates a composite image and includes an example of how such an image may be decomposed into three MRC image planes- an upper plane, a lower plane, and a selector plane;

Figure 2 contains a detailed view of a pixel map and the manner in which pixels are grouped to form blocks;

Figure 3 contains a flow chart which illustrates generally, the steps performed to practice the invention;

Figure 4 contains a detailed illustration of the manner in which blocks may be classified according to the present invention;

Figure 5 contains a detailed illustration of the manner in which blocks may be segmented based upon their classification according to the present invention;

Figure 6 contains the details of one embodiment of the manner in which block variation can be measured as required by the embodiment of the invention shown in Figure 4;

Figure 7 contains the details of an embodiment of the invention describing classification of blocks based upon the block variation measurement provided in Figure 6;

Figure 8 contains the details of an embodiment of the invention for which context may be updated based upon the block classification provided in Figure 7; and,

Figure 9 contains the details of another embodiment of the invention for updating context based upon block classification as provided in Figure 7.

[0007] The present invention is directed to a method and apparatus for separately processing the various types of data contained in a composite image. While the invention will described in a Mixed Raster Content (MRC) technique, it may be adapted for use with other methods and apparatus' and is not therefore, limited to a MRC format. The technique described herein is suitable for use in various devices required for storing or transmitting documents such as facsimile devices, image storage devices and the like, and processing of both color and grayscale black and white images are possible.

[0008] A pixel map is one in which each discrete location on the page contains a picture element or "pixel" that emits a light signal with a value that indicates the color or, in the case of gray scale documents, how light or dark the image is at that location. As those skilled in the art will appreciat , most pixel maps have values that are taken from a set of discrete, non-negative integers.

[0009] For example, in a pix l map for a color document, individual separations are often represented as digital values, often in the range 0 to 255, where 0 rep-

resents no colorant (i.e. when CMYK separations are used), or the lowest value in the range when luminanc - chrominance separations are used, and 255 represents the maximum amount of colorant or the highest value in the range. In a gray-scale pixel map this typically translates to pixel values which range from 0, for black, to 255, for the whitest tone possible. The pixel maps of concern in the currently preferred embodiment of the present invention are representations of "scanned" images. That is, images which are created by digitizing light reflected off of physical media using a digital scanner. The term bitmap is used to mean a binary pixel map in which pixels can take one of two values, 1 or 0.

[0010] Turning now to the drawings for a more detailed description of the MRC format, pixel map 10 representing a color or gray-scale document is preferably decomposed into a three plane page format as indicated in Figure 1. Pixels on pixel map 10 are preferably grouped in blocks 18 (best illustrated in Figure 2), to allow for better image processing efficiency. The docum nt format is typically comprised of an upper plane 12, a lower plane 14, and a selector plane 16. Upper plane 12 and lower plane 14 contain pixels that describe the original image data, wherein pixels in each block 18 have been separated based upon pre-defined criteria. For example, pixels that have values above a certain threshold may be placed on one plane, while those with values that are equal to or below the threshold are placed on the other plane. Selector plane 16 keeps track of every pixel in original pixel map 10 and maps all pixels to an exact spot on either upper plane 12 or lower plane 14.

[0011] The upper and lower planes are stored at the same bit depth and number of colors as the original pixel map 10, but possibly at reduced resolution. Selector plane 16 is created and stored as a bitmap. It is important to recognize that while the terms "upper" and "lower" are used to describe the planes on which data resides, it is not intended to limit the invention to any particular arrangement or configuration.

[0012] After processing, all three planes are compressed using a method suitable for the type of data residing thereon. For example, upper plane 12 and lower plane 14 may be compressed and stored using a lossy compression technique such as JPEG, while selector plane 16 is compressed and stored using a lossless compression format such as gzip or CCITT-G4. It would be apparent to one of skill in the art to compress and store the planes using other formats that are suitable for the intended use of the output document. For example, in the Color Facsimile arena, group 4 (MMR) would preferably be used for selector plane 16, since the particular compression format used must be one of the approved formats (MMR, MR, MH, JPEG, JBIG, etc.) for facsimile data transmission.

[0013] In the present invention digital imag data is pr ferably proc ssed using a MRC technique such as described above. Pixel map 10 repres nts a scanned image composed of light int nsity signals disp rs d throughout the separation at discret locations. Again, a light signal is emitted from each of these discrete locations, referred to as "picture elements," "pixels" or "pels," at an intensity level which indicat s the magnitude of the light being reflected from the original image at the corresponding location in that separation.

[0014] In typical MRC fashion, pixel map 10 must be partitioned into two planes 12 and 14. Figure 3 contains a schematic diagram, which outlines the overall process used to segment pixel map 10 into an upper plane 12 and a lower plane 14 according to the present invention. Block 18 is acquired as indicated in step 210, and is classified as indicated in step 220. In the preferred embodiment of the invention, block 18 will initially be classified as either UNIFORM, SMOOTH, WEAK_EDGE or EDGE, and its context - either TEXT or PICTURE - will be provided. The block will then be reclassified as either SMOOTH or EDGE, depending upon the initial classification and the context. Next, pixels in block 18 are segmented - placed on either upper plane 12 or lower plane 14 according to criteria that is most appropriate for the manner in which the block has been classified as indicated in step 230. This process is repeated for each block 18 in original pixel map 10 until the entire pixel map 10 has been processed. Upper plane 12, lower plane 14 and selector plane 16 are then separately compressed, using a technique that is most suitable for the type of data contained on each, as indicated in step 240.

[0015] Turing now to Figure 4, generally speaking, classification of blocks 18 into one of the four categories in step 220 as described above is preferably completed in three steps. First, the variation of pixel values within the block is determined as indicated in step 310. Block variation is best determined by using statistical measures, which will be described in detail below with reference to Figure 6. Blocks with large variations throughout are most likely to actually lie along edges of the image, while those containing little variations probably lie in uniform or at least smooth areas. Measuring the variations within the block allows an initial classification to be assigned to it as indicated in step 320. Next, image data within each block 18 is reviewed in detail to allow context information (i.e. whether the region is in the text or picture region of the image) to be updated and any necessary block re-classifications to be performed as shown in step 330. The UNIFORM blocks are reclassified as SMOOTH, and the WEAK EDGE blocks are upgraded to EDGE in a TEXT context or reclassified as SMOOTH in a PICTURE context. A smoothed version 20 of the image is also provided by applying a low pass filter to the pixel map 10. Smoothed image 20 is used in conjunction with original image data to offer additional information during classification, and also provides unscreened data for halftone regions.

[0016] Figure 5 contains d tails of the manner in which block 18 is segmented into two planes, as provided in step 230 of Figure 3. The measurement begins by

first determining at step 410 wh ther the block b ing processed has initially been classified as an EDGE in step 220. If so, the values v_p of each pixel in the block are first compared to a brightness threshold value t_s, wherein pixels that have values equal to or above t_s are viewed as "bright" pixels, while those with values below t_s are "dark" pixels. Segmenting EDGE blocks simply includes placing dark pixels on upper plane 12 as indicated in step 440, and placing bright pixels on lower plane 14 as indicated in step 450. If it is determined at step 410 that block 18 is not an EDGE, all pixels in the block are processed together, rather than on a pixel by pixel basis. Segmenting of SMOOTH (non-EDGE) pixels occurs as follows: if block 18 is in the midst of a short run of blocks that have been classified as SMOOTH, and further, all blocks in this short run are dark (v_p<t) - all data in the block is placed on upper plane 12. If the entire block 18 is substantially smooth (i.e. in a long run) or is bright (in a short run of bright pixels), all data in block 18 is placed on lower plane 14.

[0017] Turning now to Figure 6, the details of one embodim nt of the invention wherein initial block classification via block variation measurement may be accomplished as required by step 310 (Figure 4) are now described. A threshold, t_s, which allows the block to be divided into two portions is first calculated as indicated in st p 510. In the preferred embodiment of the invention, this thr shold is obtained by performing a histogram analysis on the data in the block, but many standard methods can be used to perform this analysis. For example, the value that maximizes between distances of the criteria being used for separation or provides for maximum separation between the two portions of the block can be selected. Those skilled in the art will recognize that other methods of choosing the best threshold are available and the invention is not limited to this embodiment. Block 18 is then thresholded into these two parts by comparing the light intensity value of each pixel to the selected threshold t_s, as indicated in step 520. As before, if the pixel value v_p is less than the threshold, the pixel is referred to as dark. If v_p is greater than or equal to t_s, the pixel is bright.

[0018] As stated earlier, a smooth version 20 of the image is obtained by applying a low pass filter to the original image data. Average values for bright and dark pixels are then obtained for both the original and smoothed sets of image data. Looking first at the bright pix ls, one value calculated will be v_{BPIXEL}, the average value for all of the bright pixels in original pixel map 10 (v_p ³ t_s) which are located in the area covered by block 18 as indicated in step 540. Another value, $v_{BSMOOTH}$, the average value for all of the bright pixels in smoothed version 20 of the image which are located in the area covered by block 18 will also be obtained as shown in st p 560. Dark valu s are calculated similarly. That is, v_{DPIXEL}, the average value for all of the dark pixels in original pixel map 10 (v_p < t_s) which are located in the area covered by block 18 will be obtained as shown in step 550, and $v_{DSMOOTH}$, the average value for all of the dark pixels in the smoothed version 20 of the image which are located in the area covered by block 18 will be obtained as in step 570. Once these average values are obtained, the distances d and d_s between brighter and darker averages for pixel map 10 and smoothed image 20 respectively are calculated as indicated in step 580. That is $d= V_{BPIXEL}- V_{DPIXEL}$, and $d_s = v_{BSMOOTH} - v_{DSMOOTH}$. Since d/d_s is typically almost equal to 1 for contone images, the ratio of d/d_s may be used to detect halftones.

[0019] Figure 7 contains a detailed illustration of step 320, of Figure 4, the preferred embodiment of a process for initially classifying blocks 18. As shown, a relative comparison between d and d_s is obtained as indicated in step 610 in order to determine whether the block contains contone (d » d_s) or halftone data. Block 18 will initially be classified as one of four types: UNIFORM, SMOOTH, WEAK EDGE or EDGE according to the magnitude of the distance d or d_s. Distance d is used to classify contone blocks, while distance d_s is used for halftones. For contone data d, the value from pixel map 10, is compared to value x_0 as shown in step 620.

[0020] If d is very low (i.e. d< x_0), all pixel values in the block are substantially the same and the block is classified as UNIFORM at step 640. If there are somewhat small differences in pixel values in the block such that x_0<d<x_1 as shown in step 622, the block is classified as SMOOTH, at step 650. If there are fairly large differences in pixel values in the block and x_1<d<x_2 at step 624, the block will be classified as WEAK EDGE. If the differences in the block are very large and d^3x_2 at step 624, the block will be classified as an EDGE at step 670.

[0021] If d/d_s is not approximately equal to 1, d_s is compared to threshold y_0 at step 630. It should be noted there that two different sets of thresholds are applied for halftones and contones. Thus, on most occasions, $x_0{}^1y_0$, $x_1{}^1y_1$, and $x_2{}^1y_2$. The process used to classify halftone blocks is similar to that used for contone data. Thus, if d_s<y_0 at step 630 the block is classified as UNIFORM at step 640. If y_0<d_s<y_1 in step 632, the block is classified as SMOOTH, at step 650. If y_1<d_s<y_2 as indicated in step 634, the block is classified as a WEAK EDGE at step 660. If d^3x_2 at step 634, the block will be classified as an edge at step 670.

[0022] Referring now to Figures 8 and 9, the details for updating the context of the block will now be provided. The context of a block is useful when the average between the dark and bright areas of the block is relatively high. When this is the case, the block can classified as an EDGE as long as its context is TEXT. The context is initially set equal to PICTURE. It is changed to TEXT if one of two rules is satisfied: (1) the block being proc ss d is in a long run of UNIFORM blocks and th averag of the dark pixel values in the block is great r than a preset brightness thr shold; or (2) the block has been classified as either UNIFORM, WEAK EDGE, or EDGE, one of the top, left or right neighboring blocks

has a context which has been set equal to TEXT, and the difference between that neighboring block and the current block is smaller than a preset propagation threshold.

[0023] Turning first to Figure 8, determining whether block context should be changed according to the first rule requires finding a run of blocks that have been classified as UNIFORM as indicated in step 704. Finding a run of UNIFORM blocks typically involves comparing the number of consecutive UNIFORM blocks to a run length threshold t_{LU} as indicated in step 706. The run length threshold sets the number of consecutive blocks that must be classified as UNIFORM for a run to be established. As also indicated in step 706, V_{DPIXEL}, the average value of the dark pixels for consecutive blocks is compared to the brightness threshold t_s. A large number of consecutive UNIFORM blocks with high brightness levels usually indicates that the blocks contain large background page areas (i.e. large white areas), thereby indicating that text is present. Thus, if the number of consecutive UNIFORM blocks exceeds t_{LU} and $V_{DPIXEL} > t_s$, the context for the block is changed to TEXT as indicat d in step 708.

[0024] If either the number of identified consecutive blocks is too small to establish a run or the blocks are dark (V_{DPIXEL} £ t_s), the context will remain set equal to PICTURE. Whether additional runs are present in the block will be determined as indicated in step 710, and if so the process will be repeated as indicated in the illustration.

[0025] Turning now to Figure 9, changing the context of a block to TEXT under the second rule first requires providing a propagation threshold t_p. The propagation threshold defines the level of brightness that will indicate that the block covers blank page areas. Under the second rule, the context will be changed from picture to text at step 808 if the block is not SMOOTH (i.e. is UNIFROM, and EDGE or a WEAK EDGE) as shown in step 802, either its top, left or right neighbor has a text context as indicated in step 804 and v_{BDIF}, the average difference between bright pixels in the block and bright pixels in the neighbor text context block is less than t_p as shown in step 806. Neighbor blocks are checked because presumably blocks that contain text will be located next to other blocks that contain text. However, the brightness value of the block is compared to that of its neighbor to assure that this is the case. In other words, even if the block has a neighboring block with a text context, a large difference between the average brightness of block and its neighbor means that the block contain do s not contain the large blank page areas that indicate the presence of text.

[0026] Again, the present invention is directed to segmenting the data by first id ntifying blocks that contain the edges of the image and then separating the blocks such that those which contain the smooth data and bright sides of the dges are placed on the lower plane and the dark sides of the edges are placed on the upper plane. Once each of the respective planes is generated, ordinary MRC processing continues. That is, each plane is compressed using an appropriate compression technique. In the curr ntly preferred embodiment, upper plane 12 and lower plane 14 are compressed using JPEG while the selector plane 16 is compressed using a symbol based pattern matching technique such as CCITT Group IV or a method of classifying scanned symbols into equivalence classes such as that described in US-A 5,778,095 to Davies issued July 7, 1998, the contents of which are hereby incorporated by reference. The planes are then joined together and transmitted to an output device, such as a facsimile machine or storage device.

Claims

1. A method of segmenting digital image data for mixed raster content processing, comprising:

   a) acquiring a block of the digital image data, wherein the digital image data is composed of light intensity signals in discrete locations;
   b) designating a classification for said block and providing an indication about a context of said block;
   c) segmenting said light intensity signals in said block into an upper subset and a lower subset based upon said designated classification;
   d) generating a selector set which tracks said light intensity segmentation; and
   e) separately compressing the digital image data contained in said upper and lower subsets.

2. A method of segmenting digital image data as claimed in claim 1, wherein said classification indicates that said block contains substantially smooth data and/or substantially edge data.

3. A method of segmenting digital image data as claimed in claim 1 or claim 2, wherein said classification data designating step further comprises:

   a) measuring an amount of light intensity signal variation throughout said block;
   b) assigning a classification to said block based upon said measured light intensity signal variation; and
   c) updating said context indication for said block, and designating classification for said block based upon said updated context.

4. A m thod of segmenting digital image data as claimed in any of the preceding claims, further comprising:

   a) dividing a low pass filtered version of said

block into a bright region and a dark region;
b) calculating average filtered light intensity values for said bright region and for said dark region; and
c) obtaining a difference in average filtered light intensity values between said bright region and said dark region.

5. A method of segmenting a block of digital image data into an upper and lower subset, wherein the block of data is composed of light intensity signals in discrete locations, comprising:

a) determining whether the block is located on an edge in the digital image;
b) if the block is on an edge, comparing a magnitude of each light intensity signal in the block to a brightness threshold and placing said signal in the upper subset if said light intensity magnitude exceeds said brightness threshold or in the lower subset if said light intensity magnitude is less than said brightness threshold; and
c) if the block is not located on an edge, placing the block in the upper subset if the block is in a group of blocks that have light intensity values which are indicative of smooth and dark image data, and otherwise placing the block in the lower subset.

6. A method of classifying a block of digital image data into one of a plurality of image data types, wherein the block of data is composed of light intensity signals in discrete locations, comprising:

a) dividing the block into a bright region and a dark region;
b) dividing a low pass filtered version of said block into a bright region and a dark region;
c) calculating average light intensity values for each of said bright region, said dark region, said filtered bright region and said filtered dark region; and
d) comparing a difference between said bright region and said dark region average light intensity values to a filtered difference between said bright region and said dark region average filtered light intensity values;
e) if said average light intensity difference and said average filtered light intensity difference are approximately equal finding a range of values in which said difference value falls, and classifying said block based upon said value rang ; and
f) if said average light intensity difference and said average filtered light intensity difference are not approximately equal finding a range of values in which said filtered difference value falls and classifying said block based upon said filtered value range.

7. A method according to any of claims 1 to 4, wherein blocks are classified by a method according to claim 5 or claim 6.

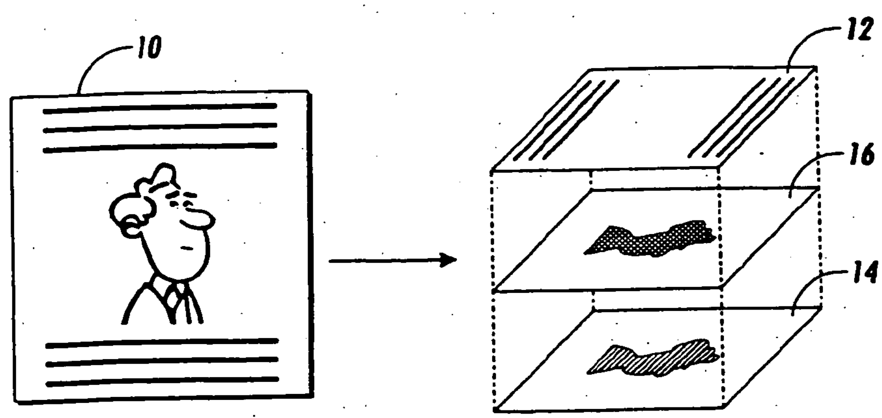

FIG. 1

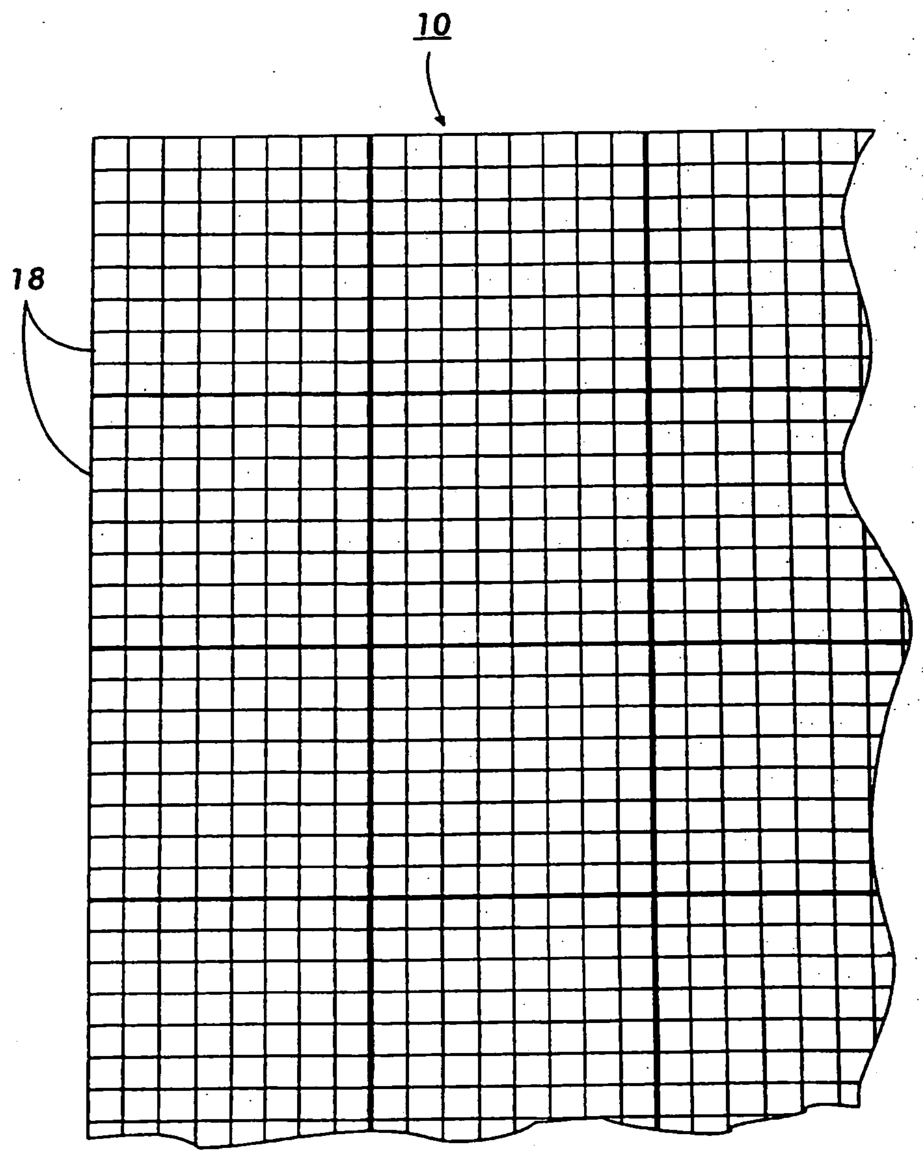

FIG. 2

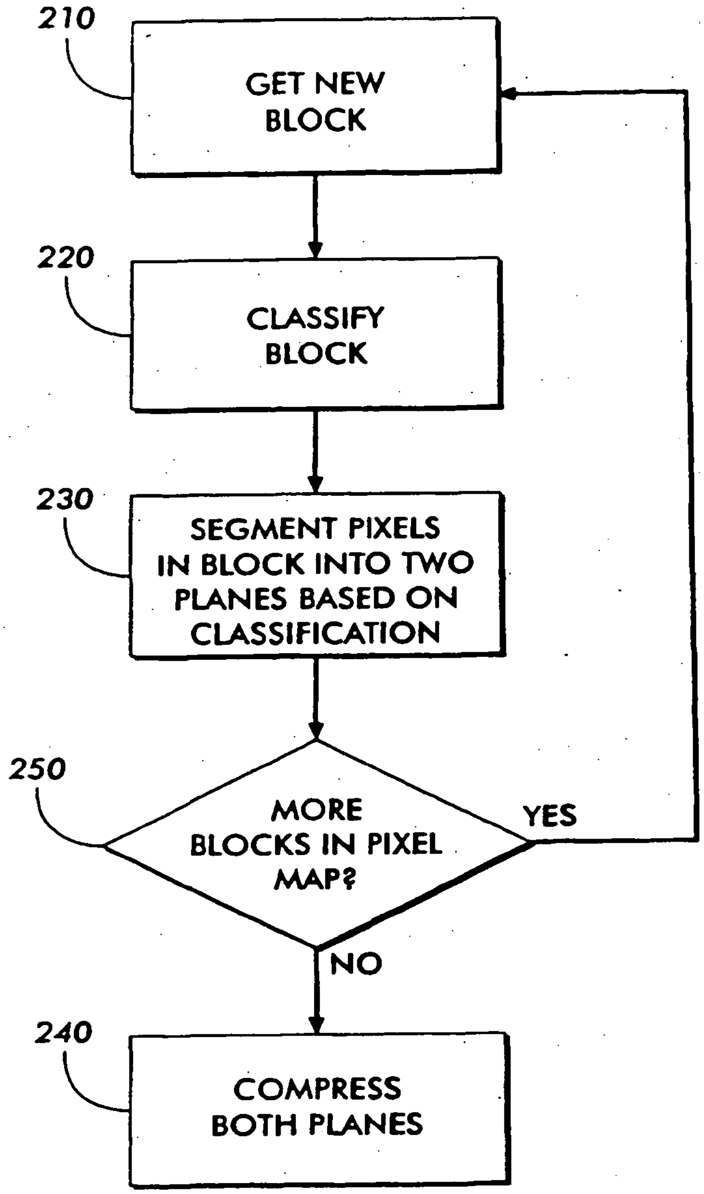
210
GET NEW BLOCK
220
CLASSIFY BLOCK
230
SEGMENT PIXELS IN BLOCK INTO TWO PLANES BASED ON CLASSIFICATION
250
MORE BLOCKS IN PIXEL MAP?
YES
NO
240
COMPRESS BOTH PLANES

FIG. 3

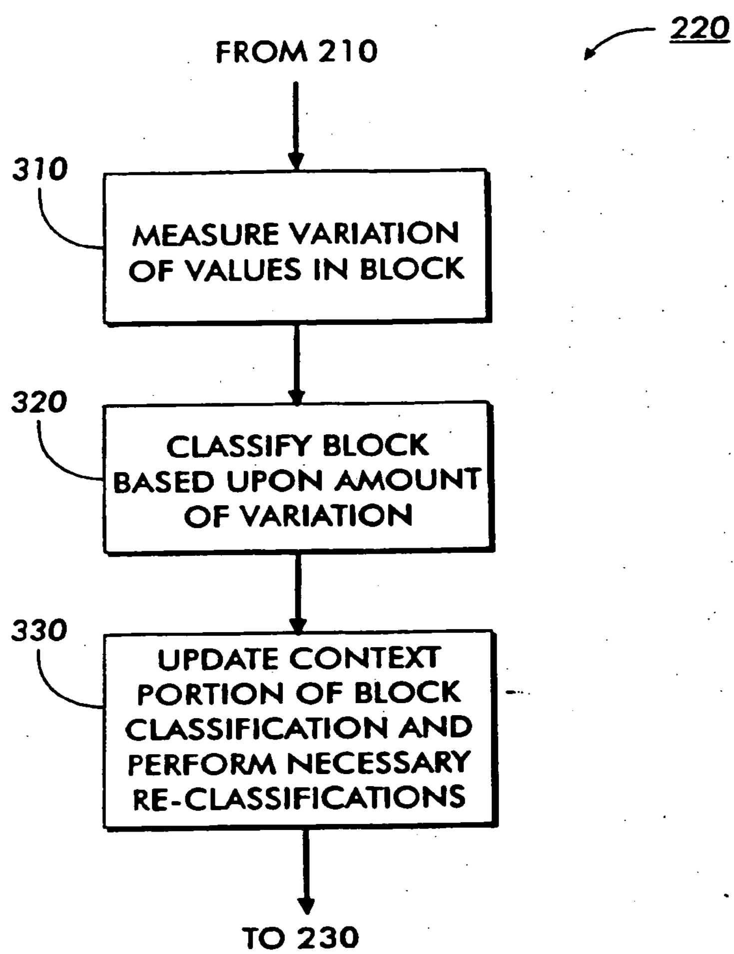
220
FROM 210
310
MEASURE VARIATION OF VALUES IN BLOCK
320
CLASSIFY BLOCK BASED UPON AMOUNT OF VARIATION
330
UPDATE CONTEXT PORTION OF BLOCK CLASSIFICATION AND PERFORM NECESSARY RE-CLASSIFICATIONS
TO 230

FIG. 4

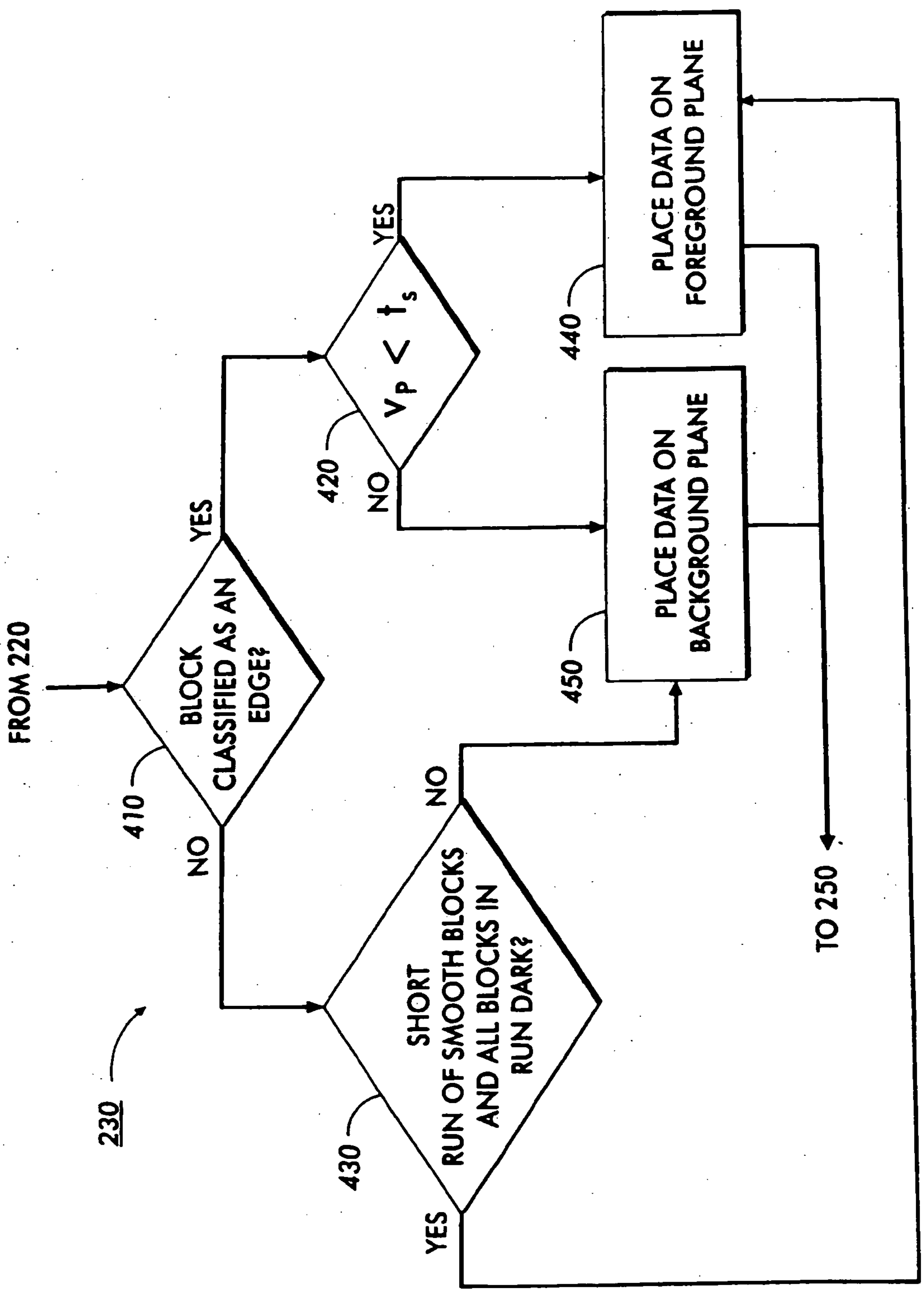
230
FROM 220
410
BLOCK CLASSIFIED AS AN EDGE?
YES
NO
420
V_P < t_s
YES
NO
440
PLACE DATA ON FOREGROUND PLANE
450
PLACE DATA ON BACKGROUND PLANE
430
SHORT RUN OF SMOOTH BLOCKS AND ALL BLOCKS IN RUN DARK?
NO
YES
TO 250

FIG. 5

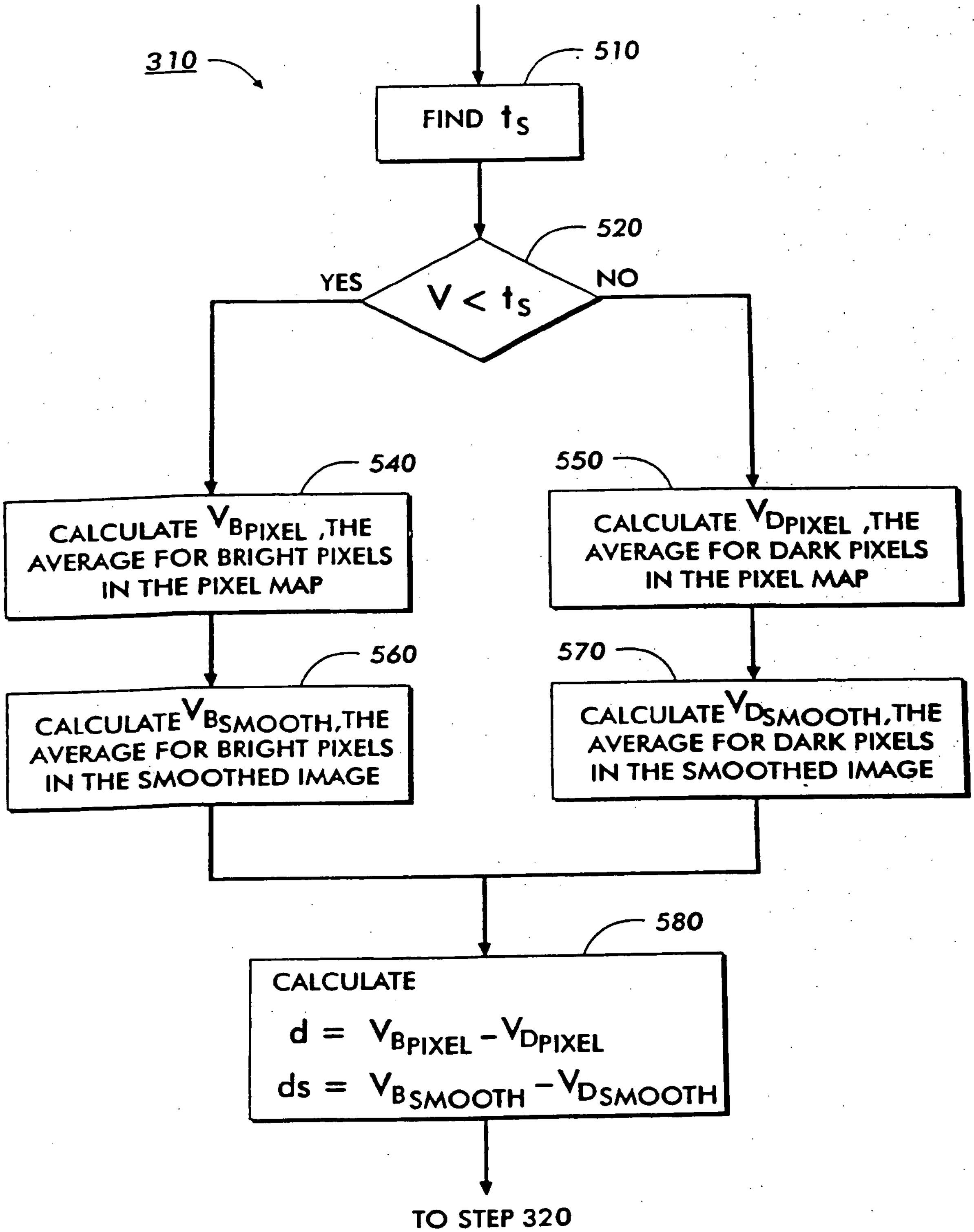
310
510
FIND t_S
520
$V < t_S$
YES
NO
540
CALCULATE V_{BPIXEL} ,THE AVERAGE FOR BRIGHT PIXELS IN THE PIXEL MAP
550
CALCULATE V_{DPIXEL} ,THE AVERAGE FOR DARK PIXELS IN THE PIXEL MAP
560
CALCULATE $V_{BSMOOTH}$,THE AVERAGE FOR BRIGHT PIXELS IN THE SMOOTHED IMAGE
570
CALCULATE $V_{DSMOOTH}$,THE AVERAGE FOR DARK PIXELS IN THE SMOOTHED IMAGE
580
CALCULATE
$d = V_{BPIXEL} - V_{DPIXEL}$
$ds = V_{BSMOOTH} - V_{DSMOOTH}$
TO STEP 320

FIG. 6

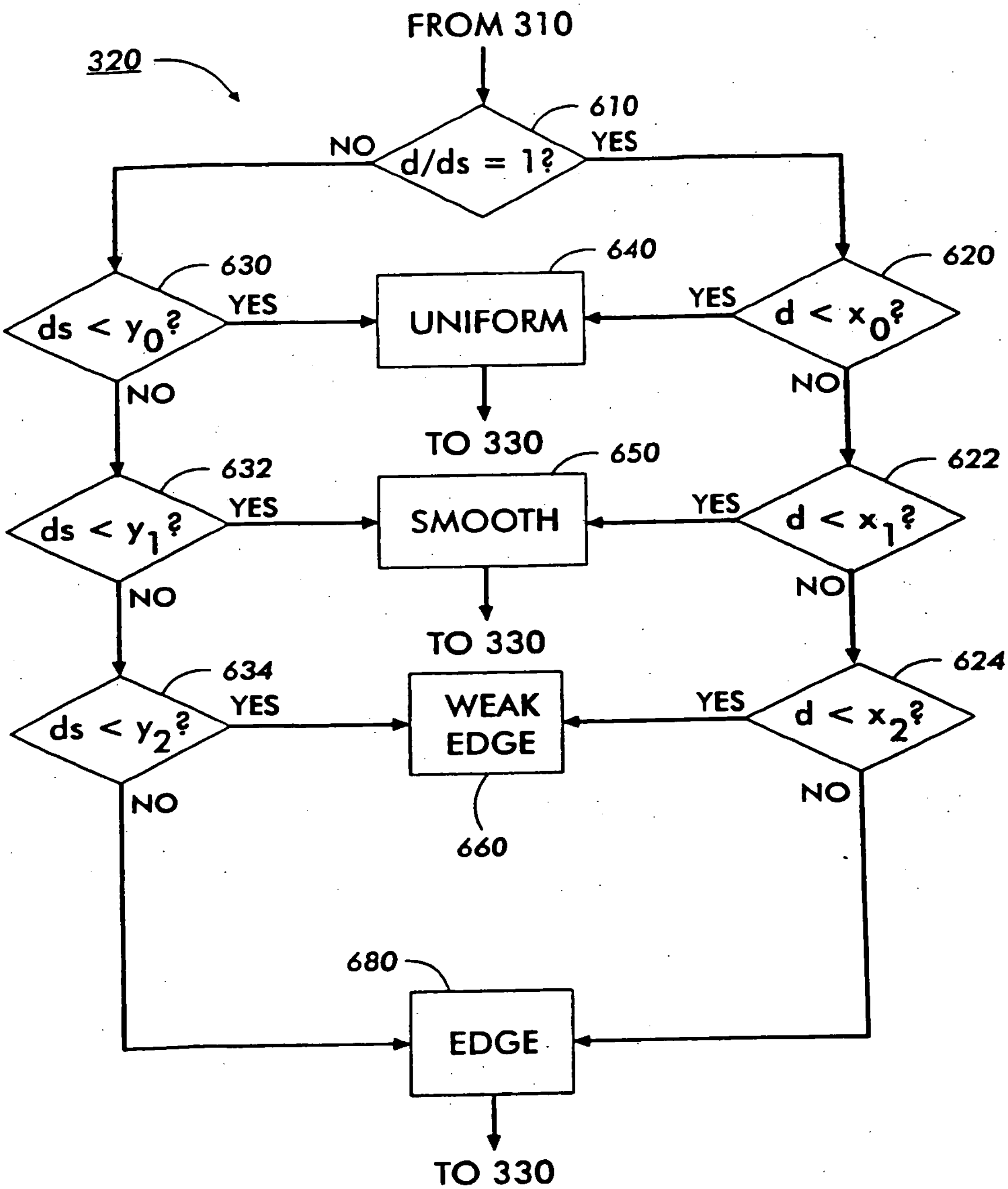
FROM 310
320
610
NO
YES
d/ds = 1?
640
630
620
YES
YES
ds < y_0?
UNIFORM
d < x_0?
NO
NO
TO 330
650
632
622
YES
YES
ds < y_1?
SMOOTH
d < x_1?
NO
NO
TO 330
634
624
YES
YES
ds < y_2?
WEAK EDGE
d < x_2?
NO
NO
660
680
EDGE
TO 330

FIG. 7

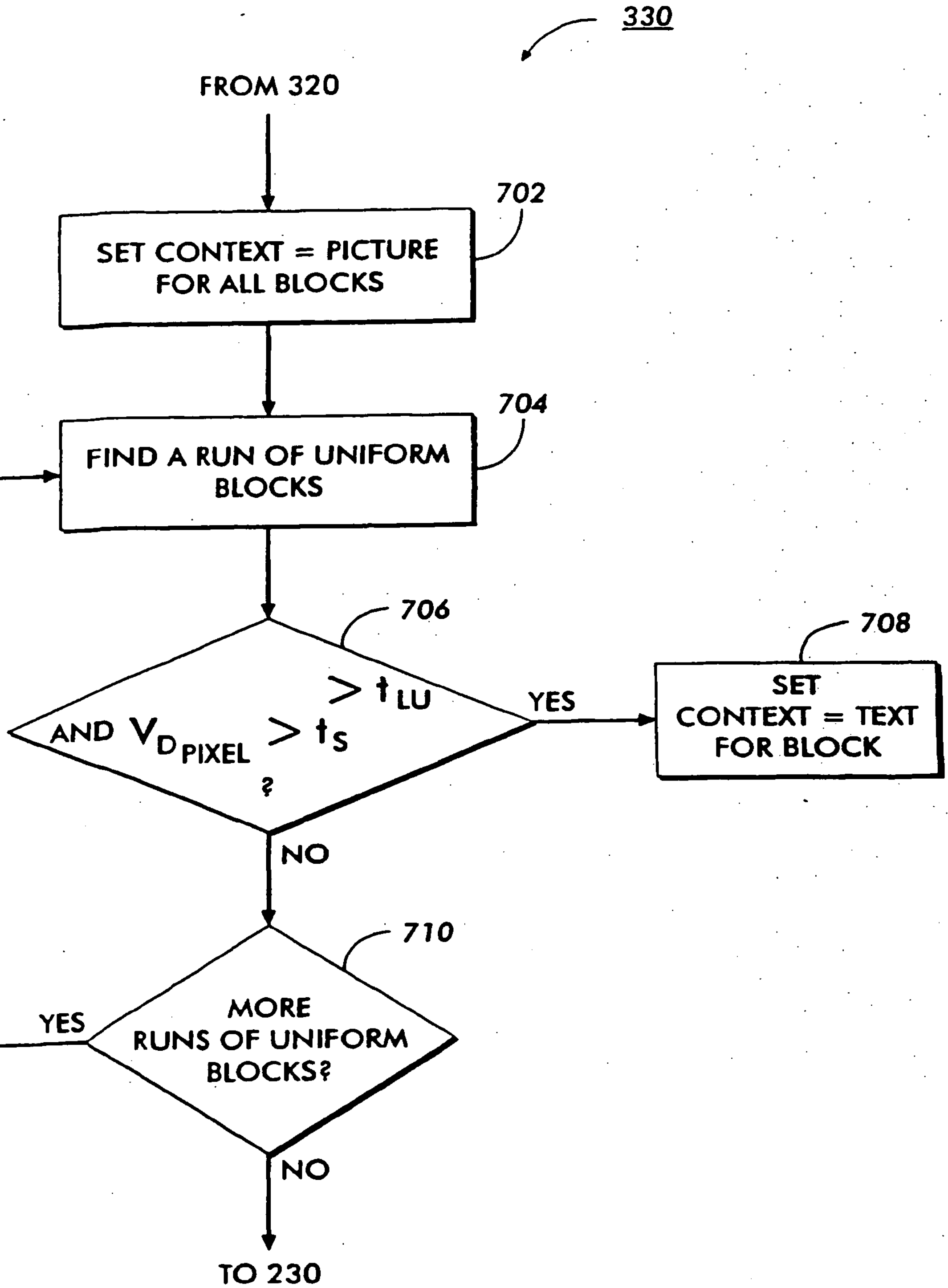
330
FROM 320
702
SET CONTEXT = PICTURE FOR ALL BLOCKS
704
FIND A RUN OF UNIFORM BLOCKS
706
> t LU
AND V D PIXEL > t S
?
YES
708
SET CONTEXT = TEXT FOR BLOCK
NO
710
MORE RUNS OF UNIFORM BLOCKS?
YES
NO
TO 230

FIG. 8

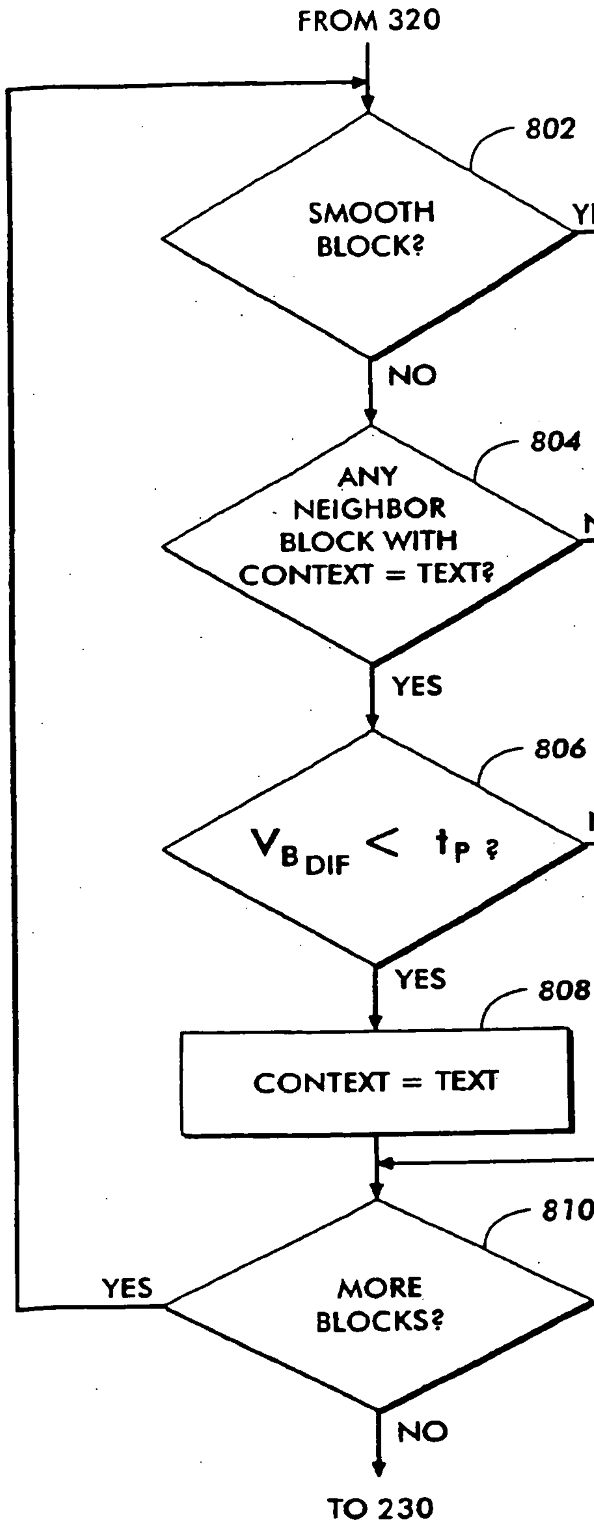
FROM 320
330
802
SMOOTH BLOCK?
YES
NO
804
ANY NEIGHBOR BLOCK WITH CONTEXT = TEXT?
NO
YES
806
$V_{B_{DIF}} < t_P$?
NO
YES
808
CONTEXT = TEXT
810
YES
MORE BLOCKS?
NO
TO 230

FIG. 9

(19) Europäisch s Patentamt
European Pat nt Offic
Office européen d s brev ts

(11) EP 1 006 716 A3

(12) EUROPEAN PATENT APPLICATION

(88) Date of publication A3:
19.09.2001 Bulletin 2001/38

(43) Date of publication A2:
07.06.2000 Bulletin 2000/23

(21) Application number: 99309522.3

(22) Date of filing: 29.11.1999

(51) Int Cl.7: **H04N 1/64**

(84) Designated Contracting States:
AT BE CH CY DE DK ES FI FR GB GR IE IT LI LU MC NL PT SE
Designated Extension States:
AL LT LV MK RO SI

(30) Priority: **02.12.1998 US 203870**

(71) Applicant: **Xerox Corporation**
R chester, New York 14644 (US)

(72) Inventors:
- **Fan, Zhigang**
  Webster, NY 14580 (US)
- **Xu, Ming**
  Rochester, NY 14618 (US)

(74) Representative: **Skone James, Robert Edmund**
GILL JENNINGS & EVERY
Broadgate House
7 Eldon Street
London EC2M 7LH (GB)

(54) Method and apparatus for segmenting data to create mixed raster content planes

(57) An improved technique for compressing a color or gray scale pixel map representing a document using an MRC format includes a method of segmenting an original pixel map into two planes (12,16), and then compressing the data or each plane in an efficient manner. The image is segmented by separating the image into two portions at the edges. One plane contains image data for the dark sides of the edges, while image data for the bright sides of the edges and the smooth portions of the image are placed on the other plane. This results in improved image compression ratios and enhanced image quality.

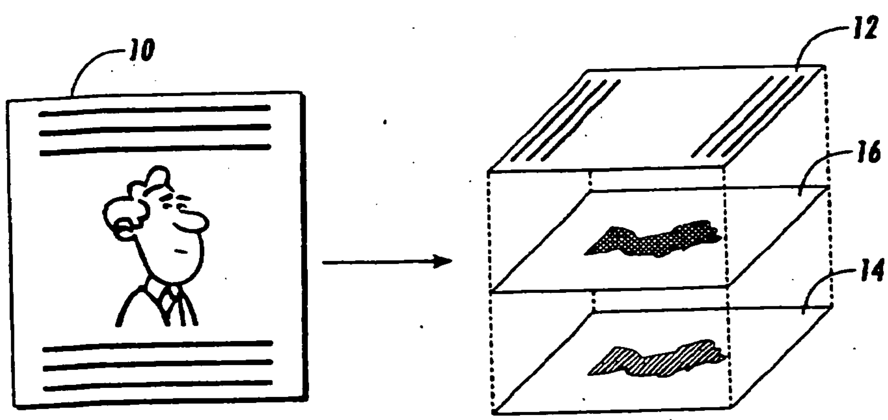

FIG. 1

EP 1 006 716 A3

European Patent Office

EUROPEAN SEARCH REPORT

Application Number
EP 99 30 9522

DOCUMENTS CONSIDERED TO BE RELEVANT

Category	Citation of document with indication, where appropriate, of relevant passages	Relevant to claim	CLASSIFICATION OF THE APPLICATION (Int.Cl.7)
A	US 5 767 978 A (FAN ZHIGANG ET AL) 16 June 1998 (1998-06-16) * abstract; claims; figures *	1,5,6	H04N1/64
A	EP 0 358 815 A (OCE NEDERLAND BV) 21 March 1990 (1990-03-21) * abstract *	1,5,6	
A	US 5 014 124 A (FUJISAWA TETSUO) 7 May 1991 (1991-05-07) * abstract *	1,5,6	
P,A	US 5 949 555 A (SAKAI AKIHIKO ET AL) 7 September 1999 (1999-09-07) * abstract *	1,5,6	
A,D	US 5 778 092 A (VINCENT LUC ET AL) 7 July 1998 (1998-07-07)		

TECHNICAL FIELDS SEARCHED (Int.Cl.7)

H04N
G06K
G06T

The present search report has been drawn up for all claims

Place of search	Date of completion of the search	Examiner
THE HAGUE	1 August 2001	Isa, S

CATEGORY OF CITED DOCUMENTS

X : particularly relevant if taken alone
Y : particularly relevant if combined with another document of the same category
A : technological background
O : non-written disclosure
P : intermediate document

T : theory or principle underlying the invention
E : earlier patent document, but published on, or after the filing date
D : document cited in the application
L : document cited for other reasons

& : member of the same patent family, corresponding document

EPO FORM 1503 03.82 (P04C01)

ANNEX TO THE EUROPEAN SEARCH REPORT ON EUROPEAN PATENT APPLICATION NO. EP 99 30 9522

This annex lists the patent family members relating to the patent documents cited in the above-mentioned European search report.
The members are as contained in the European Patent Office EDP file on
The European Patent Office is in no way liable for these particulars which are merely given for the purpose of information.

01-08-2001

Patent document cited in search report		Publication date	Patent family member(s)		Publication date
US 5767978	A	16-06-1998	EP	0855679 A	29-07-1998
EP 0358815	A	21-03-1990	DE	3881392 D	01-07-1993
			DE	3881392 T	21-10-1993
			JP	2105978 A	18-04-1990
			JP	2818448 B	30-10-1998
			US	5073953 A	17-12-1991
US 5014124	A	07-05-1991	JP	2084879 A	26-03-1990
			JP	2815157 B	27-10-1998
US 5949555	A	07-09-1999	JP	7220091 A	18-08-1995
US 5778092	A	07-07-1998	NONE		

For more details about this annex : see Official Journal of the European Patent Office, No. 12/82

EPO FORM P0459

Invited

COMPRESSION OF COMPOUND DOCUMENTS

Ricardo L. de Queiroz

Xerox Corporation
800 Phillips Rd., 128-27E, Webster, NY, 14580
queiroz@wrc.xerox.com

ABSTRACT

Compound (or mixed) document images contain graphic or textual content along with pictures. They are a very common form of documents, found in magazines, brochures, web-sites etc. Because of the very distinct nature of those two image classes (text/graphics vs. pictures), their compression invariably involves multiple compression systems and a region segmentation (classification) method. We review state-of-the-art technologies on the subject while focusing our attention on the mixed raster content (MRC) multi-layer approach. We also present new results on segmentation for MRC based on optimized rate-distortion-based block thresholding.

1. INTRODUCTION

Documents are now present in a wide spectrum of printing systems. From offset printers to home desktop computers, documents in digital form are common place. Frequently, documents are available as bitmaps and may contain text, graphics and pictures. Compound documents are images which contain a mix of textual, graphical, or pictorial contents. Those images are invariably large but a single compression algorithm that simultaneously meets the requirements for both text and image compression has been elusive. Many standard compression algorithms are available today and in common use commercially. More are continually being developed to improve on existing methods or to meet special requirements. As a rule, compression algorithms are developed with a particular image type, characteristic, and application in mind. For a different image type or application, a given algorithm either does not apply or does not perform as well as some other, better-tailored algorithm. No single algorithm is best across all image types or applications. When compressing text, it is important to preserve the edges and shapes of characters accurately to facilitate reading. Once the text is binarized, its compression is typically lossless since coding errors in text are easily perceived. The human visual system, however, works differently for typical continuous-tone images because of the richness of patterns and frequency contents. High frequency errors are better masked and lossy compression is usually employed, since lossless compression is often ineffective in this case. In terms of image resolution, text requires much higher resolution than pictures. Actually, roughly speaking, text requires few bits per pixel but many pixels per inch, while pictures require many bits per pixels but fewer pixels per inch.

Document compression is frequently linked to facsimile systems, in which large document bitmaps are compressed before transmission over telephone lines. The facsimile systems that most people are familiar with today are black-and-white (binary images) and conform to international standards set by the ITU-T (Telecommunication Standardization sector of the International Telecommunication Union, formerly known as the CCITT). These standards specify the protocols and bi-level coding procedures that sending and receiving stations use. Together with the ubiquity of the public switched telephone network (PSTN), these standards have led to the explosive growth in Group 3 black-and-white facsimile that has occurred since 1980. The same convenience and ease of use for color facsimile requires wider use of color scanners, displays and printers; faster modems and communication channels to handle the increased data volume; and equivalent standards for color facsimile. These enablers are already being put in place. For example, the ITU-T last year approved V.34 for facsimile, which supports data rates up to 33.6 Kbps, and it is now available commercially in fax machines. There is now a focus on new standards to provide color facsimile services over the PSTN and the Internet [1].

When it comes to compound documents, in order to cope with the differences between text and continuous tone images, different compression algorithms may be applied to each of the regions of the document. For that goal, some segmentation strategy has to invariably be used to discern which regions are to be encoded under which strategy. Another important parameter of a document compression system for compound documents is its imaging model. One can separate the image into different regions of interest and compress each region accordingly. In this case, the imaging model follows space segmentation where each decompressed region can be imaged into the document concurrently. Also, one can generate multiple image layers, compress each one separately and then image all the planes into one. The multilayer model will be the focus of this paper.

2. OVERVIEW

Image compression has been very intensively studied and we cannot possibly reference adequately all the most notable algorithms. However, in terms of international standards the notable algorithms for binary image compression are MH1 [2], MMR2 [3], JBIG [4] and the forthcoming JBIG-2 [5]. Multilevel compression algorithm standards are JPEG [6] and the forthcoming JPEG-2000 [7]. We assume that JPEG is the standard image compression tool while current JPEG 2000 verification model (VM) [8] is the state-of-the-art in image compression, when it comes to pictorial contents. For binary documents, MMR2 is adequate for text. JBIG can use arithmetic coding for improved performance and its multiresolution approach allows for compression of halftones. The new drive, however, in the compression of bi-level images is token-based compression. Contiguous objects are parsed and made

0-7803-5467-2/99/ $10.00 © 1999 IEEE

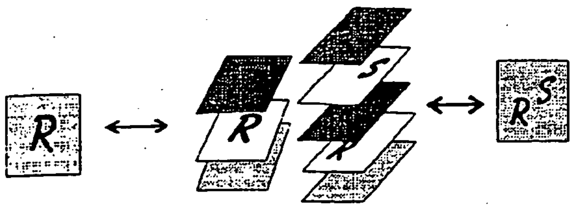

Figure 1. Illustration of MRC imaging model.

into entries in a dictionary. New objects are compared to the dictionary and if a match is found the code for that object is repeated. By making loose matches one allows the introduction of losses in exchange for higher compression. This is one of the key compression methods in document representation formats such as DigiPaper [9] and DjVu [10],[11]. Token-based compression is the heart of the forthcoming standard: JBIG-2 [5]. Even halftones can be compressed with token-based techniques by descreening the halftone and encoding new halftone patterns as objects [12]. For that, a segmentation needs to be performed to identify regions of graphics, text, halftones, etc., in a binary image, in order to improve token-based compression in JBIG-2 [13]. Other algorithms do exist which can handle graphic bitmaps well [14] and also algorithms that perform well (not optimally) for both text/graphics and pictures using non-linear filter banks [15].

Once a region is identified it can be encoded with the proper algorithm. For region identification, segmentation algorithms may be employed. For example the algorithms used in DjVu and Digipaper are already in commercial applications. Multiresolution segmentation was aplied succesfully in [16] for document compression, while [17] does the same using an approximate object location, in order to simplify the implementation. Multiscale clustering methods are also effective for segmentation [18]. We will present yet another segmentation algorithm based on block-thresholding in which the thresholds are optimized in a rate-distortion sense.

3. MIXED RASTER CONTENT

The mixed raster content (MRC) imaging model [1],[19],[20], allows for a multi-layer multi-resolution representation of a compound document. The basic 3-layer MRC model represents a color image as two color-image layers (Foreground or FG and Background or BG) and a binary image layer (Mask). The Mask layer describes how to reconstruct the final image from the FG/BG layers, i.e. to use the corresponding pixel from the FG or BG layers when the mask pixel is 1 or 0, respectively, in that position. An illustration of the imaging model is shown in Fig. 1. The foreground plane is essentially poured through the mask plane onto the background plane. The basic 3-layer model is MRC's most common form. The imaging model, however is composed of basic elementary plane pairs: FG+Mask. The FG layer is imaged onto a BG layer through the mask plane composing a new background image. Another foreground layer can be imaged onto this new background through another mask plane and the process can be repeated several times. The extended MRC model, then, allows for several planes while relying on foreground-mask pairs. A page may be represented as one, two, three or more layers, depending on its content. For example, a page consisting of a picture could use the background layer only. A page containing black-and-white text could use the mask layer, with the foreground and background layers defaulted to black and to white.

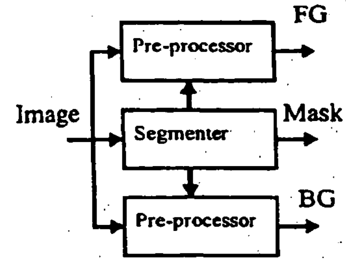

Figure 3. Diagram of a segmenter.

Once the original single-resolution image is decomposed into layers, each layer can be processed and compressed using different algorithms. The image processing operations can include a resolution change or color mapping. Layers may contain different dimensions and have offsets associated with them. If a plane contains only a small object, the effective plane can be made of a bounding box around the object. The reduced image plane is then imaged onto the larger reference plane, starting from the given offset (top, left) with given size (width, height). This avoids representing large blank areas and improves compression. The compression algorithm and resolution used for a given layer would be matched to the layer's content, allowing for improved compression while reducing distortion visibility. The compressed layers are then packaged in a format, such as TIFF-FX [21] or as an ITU-T MRC [19] data stream for delivery to the decoder. At the decoder, each plane is retrieved, decompressed, processed (which might include scaling) and the image is composed using the MRC imaging model.

MRC was originally approved for use in Group 3 color fax and is described in ITU-T Recommendation T.44. For the storage, archiving and general interchange of MRC-encoded image data, the TIFF-FX file format has been proposed [21]. TIFF-FX (TIFF for Fax eXtended) represents the coded data generated by the suite of ITU recommendations for facsimile, including single-compression methods MH, MR, MMR, JBIG and JPEG, as well as MRC. As IETF RFC 2301, TIFF-FX is a Proposed Internet Standard, currently undergoing interoperability testing. MRC has also been proposed as an architectural framework for JPEG 2000.

MRC has been used in products as DigiPaper and DjVu, whose owners built special segmenters for them, and also for check compression [22]. An analysis of the goals of the segmentation algorithm along with a better description of MRC can be found in [20]. Typical segmentation strategies are depicted in Fig. 2, which basically differ in whether one wants to move text and graphics shapes to the FG or the Mask plane. Since each layer (FG or BG) may contain unused pixels (since the pixels in that position will be selected from the other layer), those can be replaced by any color in order to enhance compression. This is the function of the pre-processor. The overall diagram is illustrated in Fig. 3. Given the pre-processors, the segmenter function is that of finding a binary mask for a given input, from which the pre-processor can derive the output layers based on the input image.

In this paper, we are interested in designing the pre-processor and segmenter for optimized compression following a basic 3-layer MRC approach. For simplicity we assume layers have same dimensions, and the encoder for FG and BG layers is JPEG. For each 8×8 input pixel block the pre-processor receives a block of equal dimensions of binary data. By inspecting the binary mask, it labels the input block pix-

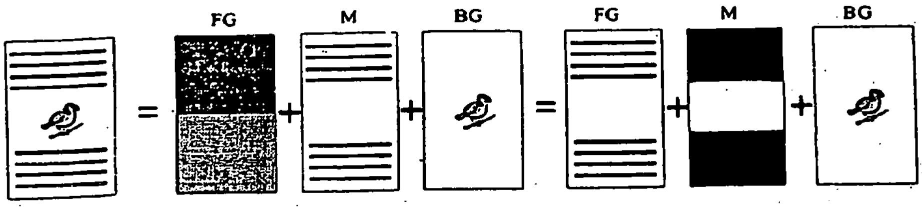

Figure 2. Illustration of typical segmentation strategies for MRC.

els as useful (U) or "don't care" (X). The X-marked pixels can be replaced by anything else since they are not going to be used for decompression. For example:

```
0 0 0 0 0 1 1 1    U U U U U X X X
0 0 0 0 0 1 1 1    U U U U U X X X
0 0 0 0 1 1 1 1    U U U U X X X X
0 0 0 0 1 1 1 1    U U U U X X X X
0 0 0 1 1 1 1 1    U U U X X X X X
0 0 1 1 1 1 1 1    U U X X X X X X
0 1 1 1 1 1 1 1    U X X X X X X X
1 1 1 1 1 1 1 1    X X X X X X X X
```

The block-wise pre-processor we used in this paper works as follows. If there are 64 X-marked pixels, the block is unused and we output a flat-block whose pixels have the average of the previous block (because of JPEG's DC DPCM [6]). If there are no X-marked pixels, the input block is output untouched. If there is a mix of U- and X-marked pixels, we follow a multi-pass algorithm: in each pass, pixels marked "X" who have at least one U-marked horizontal or vertical neighbour is replaced by the average of those neighbours and marked "U" for the next pass. The process is continued until there are no X-marked pels left in the block. The aim of the algorithm is to replace the unused parts of a block with data that will produce a smooth block based on the existing data in the U-marked pels.

4. BLOCK THRESHOLDING

Given the preprocessor just described, our goal is to find the best mapping (input block to Mask block) which will optimize compression in a rate-distortion (RD) sense. Rate is given in bits necessary to encode all 3 layers and distortion is given in MSE for the reconstruted block (after decompressing and recombining the layers). For each block, for a fix preprocessor, and without scaling, there are 2^{64} possible Mask blocks. Even if we fix the compression schemes, we cannot possibly investigate all possibilities in search for the segmentation point which yields best RD trade-off. Because of that we devised a simple preliminary experiment: to divide the mask into 16 sets of 2×2 pixels and assign each pixel in the 2×2 set the same value. The image block is also subsampled by 2 and interpolated back using nearest neighbour, so that each 2×2 group in the block has the same intensity level. Now we have only 2^{16} possible arrangements for the mask block. Sample results are shown in Fig. 4 were we plot all RD points for each given input block.

A very curious issue arises when we examine a very simple segmentation strategy: thresholding. In this, for each block a threshold is selected and the mask is found as:

$$mask_n(i,j) = u(x_n(i,j) - t_n)$$

where $x_n(i,j)$ represent the pixels at the n-th block, t_n is the correspondent threshold, and $u(k)$ is the discrete step function. Since there are 64 pixels in a block, there are at most 64 different meaningful threshold values, whereby setting t_n to be less than the darkest pixel the Mask block can be made uniform (all samples imaged from one of the layers). We then mark the RD points with squares in Fig. 4 which correspond

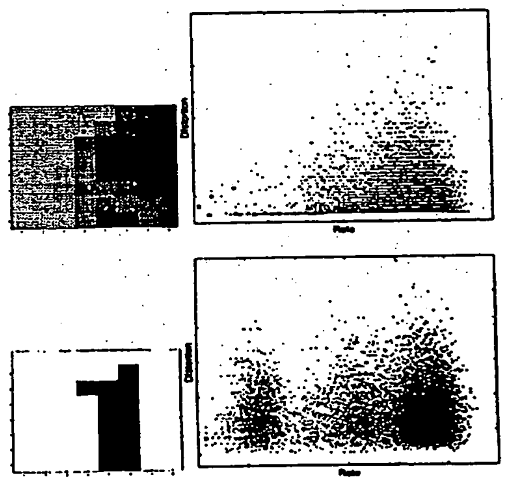

Figure 4. Sample blocks for the simplified experiment and the corresponding 64K RD plots. RD points obtained by block thresholding are marked.

to thresholding the reduced 4×4 block. It is easily seen that the mask obtained using thresholding yields among the best RD points. Although we just have shown two examples, all blocks we tested showed consistent results.

This result is not decisive but is significant. It tells us that if the results would hold for blocks of 8×8 pixels, then there is a simple way to find RD-efficient mask blocks. Note that we said RD-efficient and not optimal, since we cannot claim otherwise. Nevertheless, we pursue thresholding as a means of segmentation. The quest is to find a threshold value t_n for the n-th block. Moreover, we want to find the optimal value of t_n in an RD sense. In a block there are 64 pixels and therefore only up to 64 threshold values need to be tested. Given that the pre-processor algorithm is fixed and so are the compressors (including their parameters such as entropy coders and quantizers) every threshold value t_{kn} (k-th threshold value for the n-th block) yields a set of Mask, BG, and FG blocks, which are compressed at a total rate R_{kn} and are recombined resulting in a distortion D_{kn}. We define the cost function for a block as

$$J_n = R_{kn} + \lambda D_{kn} \ ,$$

where λ is a Lagrange multiplier which is common to all blocks. It is well known that in the optimal point all blocks operate at the same slope on the lower convex hull of the RD points. We test all t_{kn} in a block and select the one that minimizes J_n. Two examples are shown in Fig. 5, where it is shown: the input block, RD points, the RD point for mini-

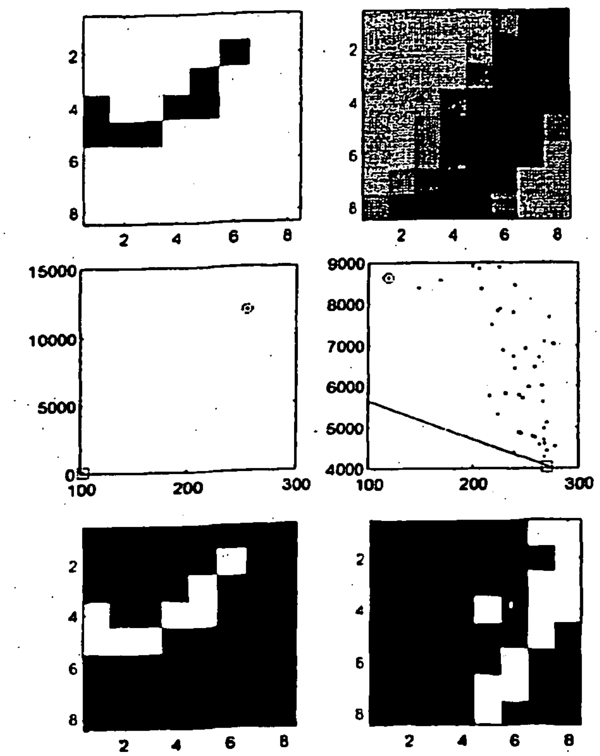

Figure 5. Sample blocks, RD plots for block thresholding and resulting Mask blocks. The operating slope is indicated along with the best RD point (□) and the RD point for a uniform mask (○).

mum J_n, the RD point for a uniform mask (no segmentation), the line with slope $-1/\lambda$ which defines the best point, and the resulting Mask block. One example is a two-tone block wherein segmentation is clearly advantageous and obvious. The other example is extracted from a picture. Note that a change in the operating point (slope of the line) may result in completely different segmentation.

The main problem in our approach is to accurately compute the rate for a given block mask. The DC term in JPEG is encoded as a function of the DC of the previous block. That forced us to use a slightly greedy approach in which we decide the operating point for a block, calculate the masks, the preprocessed layers and the JPEG compressed data based on the previous layer blocks which were already set. In this sense, results are not globally optimal. The same reason (interblock dependency) affects largely the rate of the mask plane. The rate for the mask plane is by far the largest innacuracy of the algorithm. By looking at a single block we cannot compute how many bits some transition in that mask block would cost to the overall compression. Binary compression often works with transitions and run-lengths (or tokens in the case of JBIG-2). Our simple estimate, is better correlated with the one-dimensional MH algorithm [2] although still imprecise. We simply apply a fixed penalty in bits (e.g. 7 bits) for every horizontal transition of the Mask layer. Globally, this method is a good estimator, but the hope is that it should provide at least an approximation for the sake of the RD optimization.

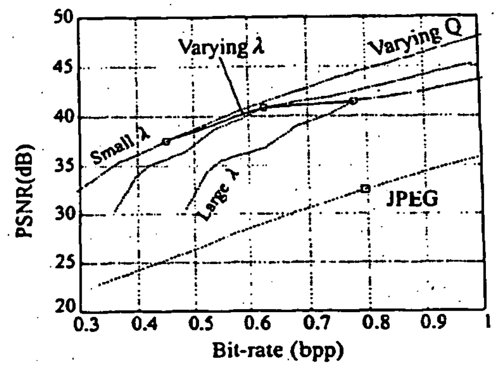

Figure 6. PSNR plots for MRC and JPEG.

PSNR plots are shown in Fig. 6 for the image "compound1" from JPEG 2000's test set. We compare MRC (using the proposed segmentation) and JPEG. The plots were obtained by scaling JPEG's example quantizer table (equal tables in both FG and BG planes) in order to vary the overall bit-rate. For the mask plane we used a simple fax MMR algorithm. The layers are then collected together using tar and gzip.

In Fig. 6, plots are shown for different values of λ, which is the operating slope of the segmenter. However, it is much more efficient to control the overall rate by modifying the compressors' parameters instead of making the Mask layer more or less complex. As λ decreases, the optimization is more biased towards minimizing rate in exchange of distortion. Nevertheless, as λ decreases the curves improve in Fig. 6. Two factors may contribute to this effect. Firstly, the innacurate calculation of the rate for the Mask layer makes it difficult to control the trade-off. The algorithm might chose to generate very complex masks since the penalty grows linearly with the number of transitions. As λ decreases, we noted that fewer portions of pictures are actually segmented. Secondly, the correlation of thresholding optimality and overall optimality may be weaker for more complex masks. In any case, results for the MRC scheme are far superior to JPEG's in terms of PSNR and can be shown to be superior to JPEG 2000's VM coder as well. A comparison of portions of an image encoded at about 0.4 bits-per-pixel (bpp) is shown in Fig. 7. It shows an MRC compressed image using: segmentation through block thresholding for very small λ; JPEG compression for both FG and BG layers; and CCITT's MMR for the Mask layer. It also shows the result using JPEG and the actual Mask plane used for MRC. Other images and comparisons can be shown but space limitations preclude the presentation of more results.

5. REMARKS

Optimized block thresholding seems to be an effective way to segment a compound document image for compression. If the complexity is not acceptable for a given application, one can use this procedure to guide and train non-RD-based segmentations strategies. Results so far are not decisive. Further efforts will be concentrated on better methods to estimate the rate achieved by compressing the Mask layer and investigating the reasons why minimization of rate is much more important than minimization of distortion, in the segmenta-

tion algorithm. Further results and details will be presented in a forthcoming full paper [23].

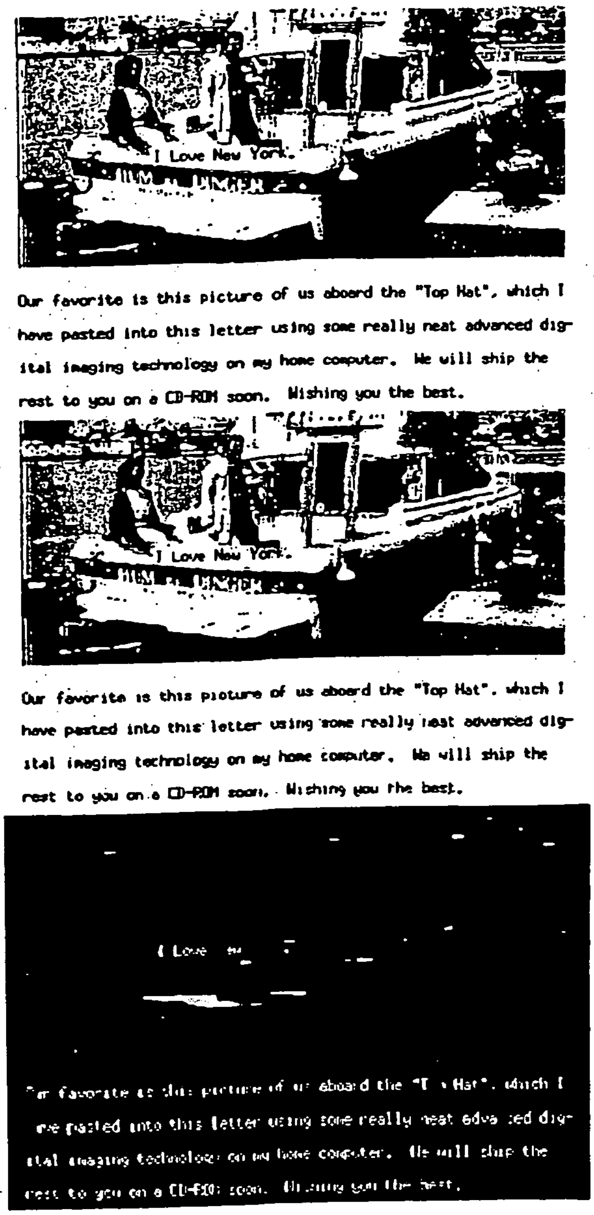

Figure 7. Top: portion of a reconstructed image after compression using MRC at 0.37bpp and 35.4dB PSNR. Middle: same for JPEG at 0.39bpp and 23.9dB PSNR. Bottom: mask used for segmentation.

REFERENCES

[1] R. Buckley, D. Venable and L. McIntyre, "New developments in color facsimile and internet fax," *Proc. of IS&T's Fifth Color Imaging Conference*, pp. 296–300, Scottsdale, AZ, Nov. 1997.

[2] ITU-T Rec. T.4, Standardization of group 3 facsimile apparatus for document transmission, July 1996.

[3] ITU-T Rec. T.6, Facsimile coding schemes and coding control functions for group 4 facsimile apparatus, November 1988.

[4] ITU-T Rec. T.82, Information technology - Coded representation of picture and audio information - Progressive bi-level image compression, March 1995.

[5] JBIG2 Working Draft WD14492, ISO/IEC JTC1/SC29 JBIG Committee, 21 August 1998,

[6] W. P. Pennebaker and J. L. Mitchell, *JPEG: Still Image Compression Standard*, Van Nostrand-Reinhold, 1993.

[7] ISO/IEC JTC1/SC29 WG1, JPEG 2000 Committee, Working Draft 2.0, June 25, 1999.

[8] ISO/IEC JTC1/SC29 WG1, JPEG 2000 Committee, JPEG 2000 Verification Model (Technical Description), April 22, 1999.

[9] D. Huttenlocher and W. Rucklidge, "DigiPaper: a versatile color document image representation," in this Proceedings.

[10] L. Bottou, P. Haffner, P. Howard, P. Simard, Y. Bengio and Y. LeCun, "High quality document image compression using DjVu," *Journal of Electronic Imaging*, 7(3), pp. 410-425. July 1998.

[11] L. Bottou, P. Haffner, P. Howard and Y. LeCun, "Color documents on the Web with DjVu," in this Proc.

[12] M. Valliappan, B. Evans, D. Tompkins, F. Kossentini, "Lossy compression of stochastic halftones with JBIG2," in this Proceedings.

[13] D. Tompkins and F. Kossentini, "A fast segmentation algorithm for bi-Level image compression using JBIG2," in this Proceedings.

[14] Y. Yoo, Y. Kwon, A. Ortega, "Embedded image domain compression for simple images," *Proc. 32nd Asilomar Conf. on Sig. Sys. Comp.*, Pacific Grove, CA, Nov. 1998.

[15] R. L. de Queiroz, D. A. Florencio and R. W. Schafer , "Non-expansive pyramid for image coding using nonlinear filter banks," *IEEE Trans. on Image Processing*, Vol. 7, pp. 246-252, Feb. 1998.

[16] H. Cheng and C. Bouman, "Document compression based on multiscale segmentation," in this Proc.

[17] A. Said and A. Drukarev, "Simplified segmentation for compound image compression," in this Proceedings.

[18] D. Mukherjee, S. Acton, "Document page segmentation using multiscale clustering," in this Proceedings.

[19] Draft Recommendation T.44, Mixed Raster Content (MRC), ITU-T Study Group 8, Question 5, May 1997.

[20] R. de Queiroz, R. Buckley and M. Xu, "Mixed raster content (MRC) model for compound image compression," *Proc. EI'99, VCIP*, SPIE Vol. 3653, pp. 1106-1117, Feb. 1999.

[21] IETF RFC 2301. File Format for Internet Fax. March 1998. ftp://ftp.isi.edu/in-notes/rfc2301.txt.

[22] J. Huang, Y. Wang and E. Wong, "Check image compression using a layered coding method," *Journal of Electronic Imaging*, 7(3), pp. 426-442, July 1998.

[23] R. de Queiroz, T. Tran and Z. Fan, "Optimizing block-thresholding segmentation for multi-layer compression of compound images," submitted to *IEEE Trans. Image Proc*, 1999.